In the Claims

1. (Currently Amended) A method for writing data on a data storage device, comprising:

said data storage device receiving a write command;

obtaining a starting logical block addresses (LBA) ~~LBA~~ and a LBA transfer length from said write command;

obtaining a first write once ready many (WORM) ~~WORM~~ pointer from a WORM pointer memory, said WORM pointer providing an inventory of LBAs where WORM data can be written within said data storage media; and

in response to said starting LBA being greater than or equal to said WORM pointer, executing said write command.

2. (Original) The method of claim 1, wherein executing said write command writes said data as WORM data on said data storage device.

3. (Original) The method of claim 1, further comprising:

in response to said starting LBA being less than said first WORM pointer, aborting said write command.

4. (Original) The method of claim 1, further comprising:

in response to determining that said write command executed without errors:

calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length; and

storing said second WORM pointer in said WORM pointer memory.

5. (Original) The method of claim 1, further comprising:

in response to determining that said write command executed without errors:

calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length;

storing said second WORM pointer in said WORM pointer memory; and storing a date stamp for each said WORM pointer stored in said WORM pointer memory.

6. (Original) The method of claim 1, further comprising:
in response to determining that said write command executed with at least one error, rewriting said data.

7. (Original) The method of claim 1, further comprising:
in response to determining that said write command executed with at least one error, rewriting said data beginning at said starting LBA.

8. (Original) The method of claim 1, further comprising:
in response to determining that said write command executed with at least one error, rewriting said data beginning at an LBA that is greater than said starting LBA.

9. (Original) The method of claim 1, further comprising:
in response to receiving a first inquiry command from a host computer, sending a device type to said host computer.

10. (Original) The method of claim 9, further comprising:
in response to receiving a second inquiry command from said host computer, sending a worm pointer to said host computer.

11. (Cancelled)

12. (Cancelled)

13. (Currently Amended) A data storage device, comprising:
a data storage media for storage of data;
a processor for controlling said data storage device;

a write once ready many (WORM) ~~WORM~~ pointer memory coupled to said processor for storage of a WORM pointer, said WORM pointer providing an inventory of locations where WORM data can be written within said data storage media;

a host device interface coupled to said processor for receiving commands from a host computer.

14. (Original) The data storage device of claim 13, wherein said data is stored as WORM data on said data storage media.

15. (Currently Amended) The data storage device of claim 13, wherein said processor obtains a starting logical block addresses (LBA) ~~LBA~~ and a LBA transfer length from a write command received by said host device interface, obtains a first WORM pointer from said WORM pointer memory and in response to said starting LBA being greater than or equal to said first WORM pointer, executes said write command.

16. (Original) The data storage device of claim 13, wherein said WORM pointer memory is an EPROM.

17. (Original) The data storage device of claim 13, wherein said WORM pointer memory is a PROM.

18. (Original) The data storage device of claim 13, wherein said WORM pointer memory is a FLASH memory.

19. (Original) The data storage device of claim 13, wherein said WORM pointer memory is located inside a sealed portion said data storage device.

20. (Original) The data storage device of claim 13, further comprising:

a memory device for storage of a date stamp associated with each said WORM pointer.

21. (Currently Amended) A data storage system, comprising:

a host computer comprising a data storage device interface;

a data storage device comprising:

a data storage media for storage of data;

a processor for controlling said data storage device;

a write once ready many (WORM) ~~WORM~~ pointer memory coupled to said processor for storage of a WORM pointer, said WORM pointer providing an inventory of locations where WORM data can be written within said data storage media;

a host device interface coupled to said processor for sending and receiving commands with respect to said host computer.

22. (Original) The data storage system of claim 21, wherein said data is stored as WORM data on said data storage media.

23. (Original) The data storage system of claim 21, wherein said processor obtains a starting LBA from a write command received by said host device interface, obtains a first WORM pointer from said WORM pointer memory and in response to said starting LBA being greater than or equal to said WORM pointer, executes said write command.

24. (Currently Amended) An article of manufacture comprising a ~~data storage~~ computer readable medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform method steps for writing data on a data storage device, said steps comprising:

said data storage device receiving a write command;

obtaining a starting logical block addresses (LBA) ~~LBA~~ and a LBA transfer length from said write command;

obtaining a first write once ready many (WORM) ~~WORM~~ pointer from a WORM pointer memory, said WORM pointer providing an inventory of LBAs where WORM data can be written within said data storage media; and

in response to said starting LBA being greater than or equal to said WORM pointer, executing said write command.

25. (Original) The article of manufacture of claim 24, wherein executing said write

command writes said data as WORM data on said data storage device.

26. (Original) The article of manufacture of claim 24, wherein said method steps further comprises: in response to said starting LBA being less than said first WORM pointer, aborting said write command.

27. (Original) The article of manufacture of claim 24, wherein said method steps further comprises:

in response to determining that said write command executed without errors:

calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length; and

storing said second WORM pointer in said WORM pointer memory.

28. (Original) The article of manufacture of claim 24, wherein said method steps further comprises:

in response to determining that said write command executed without errors:

calculating a second WORM pointer which is equal to the numerical sum of said first WORM pointer and said transfer length;

storing said second WORM pointer in said WORM pointer memory; and

storing a date stamp for each said WORM pointer stored in said WORM pointer memory.

29. (Original) The article of manufacture of claim 24, wherein said method steps further comprises:

in response to determining that said write command executed with at least one error, rewriting said data.

30. (Original) he article of manufacture of claim 24, wherein said method steps further comprises:

in response to determining that said write command executed with at least one error, rewriting said data beginning at an LBA that is greater than or equal to said starting LBA.

31. (Original) The article of manufacture of claim 24, wherein said method steps further comprises:

in response to receiving a first inquiry command from a host computer, sending a device type to said host computer.

32. (Original) The article of manufacture of claim 31, wherein said method steps further comprises:

in response to receiving a second inquiry command from said host computer, sending a worm pointer to said host computer.

33. (Cancelled)

34. (Cancelled)